UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/418,943	10/15/1999	TIMOTHY CHARLES SOWELL	202231	9508

7590 08/10/2007

MARK JOY
LEYDIG VOIT & MAYER LTD
TWO PRUDENTIAL PLAZA SUITE 4900
180 NORTH STETSON
CHICAGO, IL 606016780

EXAMINER	
NGUYEN, NGA B	
ART UNIT	**PAPER NUMBER**
3692	

MAIL DATE	DELIVERY MODE
08/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/418,943
Filing Date: October 15, 1999
Appellant: SOWELL, TIMOTHY CHARLES

Mark Joy, Reg. No. 35,562
For Appellant

EXAMINER'S ANSWER

This is in response to the Appeal Brief filed on December 7, 2004 appealing from the Office action mailed on July 7, 2004.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The statement of the status of Amendments contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of invention contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The summary of the ground of rejection to be reviewed on appeal contained in the brief is correct.

(7) Claim Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence relied Upon

5,825,883	Archibald et al.	10-1998
6,363,486	Knapton, III	03-2002
6,499,035	Sobeski	12-2002

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, 53-55, and 74-77 are rejected under 35 U.S.C. 103(a) as being unpatentable over Archibald et al (hereinafter Archibald), U.S. Patent No. 5,825,883, in view of Knapton, III (hereinafter Knapton), U.S. Patent No. 6,363,486, and further in view of Sobeski, U.S. Patent No. 6,499,035.

Regarding claim 1, Archibald discloses a method for charging customers for use of software comprising the steps of:

establishing a use-based pricing scheme for a set of software modules (column 7, lines 53-67);

distributing the set of software modules to a customer (column 4, lines 20-50);

monitoring customer use of the software modules (column 4, lines 50-65); and

charging the customer according to use of the distributed software modules as determined during the monitoring step (column 4, lines 50-60).

Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer

computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable.

Regarding claim 2, Archibald in modified by Knapton and Sobiski further disclose the customer creates a number of instances from a software module, and use of the software module is measured according to instances detected at a site of the customer during the monitoring step (Archibald, column 6, lines 48-60; the meter software module generates accounting information includes many instances the user information, user ID, user account number, etc, if the Archibald's software application is developed using an object-oriented programming such as Sobiski's Java objects, the user information, user ID, user account number, etc in Archibald are objects or instances executed on the customer's computer).

Regarding claim 3, Archibald in modified by Knapton and Sobiski further disclose instances created from a software module are periodically accessed to determine use during the monitoring step (the meter software module accumulates the accounting information for a predetermined period of time, column 12, lines 33-42).

Regarding claim 5, Archibald in modified by Knapton and Sobiski further disclose providing a demonstration mode for instances such that instances in the demonstration mode are executable at a customer site without charge (demonstration mode is "trial use field 166", column 9, lines 25-35).

Regarding claim 6, Archibald in modified by Knapton and Sobiski further disclose maintaining a single agreement governing use of instances created from the set of software modules for an enterprise (column 7, lines 40-67; Many of instances are created when the user has terminated use of the software module, i.e. an application identification field, a price filed, a user unit field, an amount user field, etc).

Regarding claim 7, Archibald in modified by Knapton and Sobiski further disclose monitoring a termination date for instances derived from a software module having a time-limited duration (instance "the rent to own field", column 9, lines 19-25).

Regarding claim 9, Archibald in modified by Knapton and Sobiski further disclose the steps of: maintaining an account for storing units of credit for a customer; and wherein charging step comprises decrementing the customer's credit account by an appropriate number of units of credit based upon monitoring step (column 7, lines 20-30).

Regarding claim 10, Archibald in modified by Knapton and Sobiski further disclose generating a report summarizing use of software modules at the customer site (column 7, lines 40-67).

Regarding claim 11, Archibald in modified by Knapton and Sobiski further disclose the charging step is based upon registered uses of a software module (column 7, lines 53-67).

Regarding claims 12 and 74, Archibald in modified by Knapton and Sobiski further disclose the registered uses of a software module are measured according to

execution and creation of an instance from the software module (see claim 1 above for details).

Regarding claims 14-15, Archibald in modified by Knapton and Sobiski further disclose the software module is an object class for creating an application engine object or an object class for creating a view engine object (see claim 1 above for details and column 5, line 65-column 6, line 47, the software module is download to the meter module at the consumer computer).

Regarding claim 16, Archibald in modified by Knapton and Sobiski further disclose the monitoring step comprises determining a time duration that an object instantiated from a software module is active (column 8, lines 43-46).

Regarding claim 17, Archibald in modified by Knapton and Sobiski further disclose the monitoring step comprises registering execution of an instance that tracks throughput of a process (The embedded check data 76 is utilized by the meter module 26 to insure that during operation, the program is being executed as the publisher had intended, column 5, line 65-column 6, line 12).

Regarding claim 18, Archibald in modified by Knapton and Sobiski further disclose individual ones of the set of software modules are individually priced (column 7, lines 52-67).

Regarding claim 19, Archibald in modified by Knapton and Sobiski further disclose the set of software modules includes at least a first software module supplied by a third party vendor and further comprising the step of: compensating a third party vendor based upon a use by a customer of the first software module determined during the monitoring step (column 7, lines 20-30).

Regarding claims 20-21, Archibald in modified by Knapton and Sobiski further disclose the distributing step comprises transmitting the set of software modules via a network connection comprises an Internet connection (column 3, lines 64-65).

Regarding claim 22, Archibald in modified by Knapton and Sobiski further disclose a step of reporting usage information to a software brokerage facility (column 10, lines 33-45).

Regarding claim 23, Archibald in modified by Knapton and Sobiski further disclose the reporting step includes identifying the location of an instance created from a software module (an application identification field identifies which software module included in the particular combined digital application module 212, see figure 5 and column 10, lines 1-12).

Regarding claim 24, Archibald in modified by Knapton and Sobiski further disclose determining that a license manager has not reported to a software brokerage facility and in response registering a communication failure at a central licensing facility (column 19, lines 40-50).

Regarding claim 25, Archibald in modified by Knapton and Sobiski further disclose the monitoring step includes storing use information in summary format in a database (column 7, lines 40-67).

Regarding claims 27-28, Archibald in modified by Knapton and Sobiski further disclose the software modules relate to industrial manufacturing automation software or relate to industrial manufacturing information software (automation software: active application, information software: passive application column 2, lines 37-44).

Regarding claim 29, Archibald in modified by Knapton and Sobiski further disclose maintaining an agreement governing use of instances created from the set of software modules for an enterprise wherein the instances comprise both lifetime billed

and use-based billed instances (column 7, lines 53-67 and column 9, lines 19-25 and see claim 1 for details).

Regarding claims 30, Knapton discloses providing configuration tools enabling a user to create customized instances from the software modules (column 6, lines 48-67). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to include this feature in Archibald's for the purpose of helping the user to create customized instances based on the configuration tools in the software product.

Regarding claim 31, Archibald discloses a method for vending software in the form of software modules via electronic commerce channels comprising the steps of:

maintaining an electronic commerce site including a software module selection interface, the software module selection interface enabling a customer to request a software module for use at a customer site (column 4, lines 20-50);

providing a software module management framework to the customer for installation at a customer site, wherein the management framework includes components for registering use of the software module at the customer site (column 5, line 65-column 6, line 47); and

charging the customer based upon registered use of the software module (column 6, lines 48-67).

Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the

software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable.

Regarding claim 32, Archibald in modified by Knapton and Sobiski further disclose the use of the software module comprises executing an instance created from the software module (column 6, lines 33-47).

Regarding claim 33, Archibald in modified by Knapton and Sobiski further disclose the use of the software module comprises creating an instance from the downloadable module (column 4, lines 30-32).

Regarding claim 34, Archibald f in modified by Knapton and Sobiski urther disclose registering use of the downloadable module provides a measure of throughput of an industrial process (The embedded check data 76 is utilized by the meter module 26 to insure that during operation, the program is being executed as the publisher had intended, column 5, line 65-column 6, line 12).

Regarding claim 35, Archibald in modified by Knapton and Sobiski further disclose the module management framework supports creation of instances from software modules at the customer site having deferring use modes including at least: a lifetime mode and a use-based mode, and wherein method comprises the further step

of registering execution of instances operating in the use-based mode (column 9, lines 19-25 and column 7, lines 40-67).

Regarding claim 37, Archibald discloses a method for charging customers for use of software comprising the steps of:

providing a set of individually identifiable units of executable software (column 7, lines 58-67;

individually pricing ones of the set of individually identifiable units of executable software (column 7, lines 58-67);

authorizing use of the executable software (column 6, lines 33-47); and

charging a customer based upon use of selected ones of the set of individually identifiable units of executable software (column 7, lines 58-67).

Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of

time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable.

Regarding claim 38, Archibald in modified by Knapton and Sobiski further disclose the authorizing step comprises transmitting a license file containing code enabling use by the customer of the executable software (column 6, lines 33-47 and column 11, line 28-column 12, line 22).

Regarding claim 39, Archibald in modified by Knapton and Sobiski further disclose the steps of: integrating self-monitoring process software within the executable software; and registering use of the executable software by the self-monitoring process (column 4, lines 1-6 and column 5, line 65-column 6, line 12).

Regarding claim 40, Archibald in modified by Knapton and Sobiski further disclose the executable software is industrial automation software (active application, column 2, lines 38-44).

Regarding claim 42, Archibald discloses a method for charging customers for user of software comprising the steps of:

first providing a set of software modules for software customers (column 4, lines 20-35);

second providing a software licensing facility including a broker facility through which software customers pay for software execution units, and wherein the broker facility includes a set of software customer accounts (column 4, lines 20-35 and column 7, lines 20-30); and

charging a software customer account a number of software execution value units based upon the value of software modules utilized by a customer (column 7, lines 53-67).

Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable.

Regarding claim 43, Archibald in modified by Knapton and Sobiski further disclose the charging step is performed by an automated billing process (column 7, lines 20-30).

Regarding claim 44, Archibald in modified by Knapton and Sobiski disclose providing an on-line customer interface; and wherein the first providing step includes the step of providing a network interface enabling users to download software modules from a remote location (column 3, lines 64-65 and column 4, lines 30-32).

Regarding claims 53-55, Archibald discloses a software module facilitating automated distribution of software to customers comprising: a supplier identification; a

product description; an executable program; a billing definition includes a usage rate and a lifetime rate (column 6, lines 33-60).

Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable.

Claims 75-77 have similar limitations as found in claims 32-34, therefore, are rejected by the same rationale.

3. Claims 4, 8, 26, 36, 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Archibald et al (hereinafter Archibald), U.S. Patent No. 5,825,883, in view of Knapton, III (hereinafter Knapton), U.S. Patent No. 6,363,486, in view of Sobeski, U.S. Patent No. 6,499,035, and further in view of Ahmad, U.S. Patent No. 5,925,127.

Regarding claims 4 and 36, Archibald does not teach the monitoring step comprises registering each day that an instance created from a software module is active and wherein the charging step comprises charging the customer a daily rate for use of the software module. However, Ahmad teaches charging the customer a daily rate for use of the software module (column 8, lines 65-67 and column 1, lines 50-51). Therefore, it is obvious to include charging the customer a daily rate for use of the software module in Archibald's for the purpose of time consuming.

Regarding claims 8, 26, and 41, Archibald does not explicitly teach the self-monitoring process software comprises includes functions for issuing a warning in response to detecting an upcoming expiration date for an instance of a software module or informing the customer of a need to reorder credit to continue using the executable software. However, Ahmad discloses the self-monitoring process software comprises includes functions for issuing a warning in response to detecting an upcoming expiration date for an instance of a software module or informing the customer of a need to reorder credit to continue using the executable software (column 14, line 28-column 15, line 5). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to include these features in Archibald's for the purpose of reminding the customer the expired date of software, so the customer does not forget to re-order credit in order to continue to use the software.

(10) Response to Argument

In response to the appellant's argument regarding to Group I: Claims 1, 9-11, 18-22, 25, 29, 31, 35, 37, 39, 42, 43, and 44, examiner submit that Archibald does not disclose the set of software modules comprise at least one object class from which objects are instantiated on a customer system and software usage is measured

according to object instances created from the at least one object class. However, Knapton discloses the software modules executed on the customer computer system is developed using object oriented programming comprises C++, Java, Visual Basic, COM, ActiveX, etc...(column 3, lines 30-58). Sobiski discloses the software modules executed on the customer computer system is developed using Java objects comprise at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class (column 6, line 47-coumn 7, line 60). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to modify Archibald's to include the feature above for the purpose of developing the Archibald's software application using object-oriented programming language such as Java objects as taught by Knapton and Sobiski for the purpose of time consuming and cost saving because object oriented programming is easy to create, cost effective to modify, and reusable. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and In re Jones, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In response to the appellant's argument regarding to Group II: Claim 2, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the customer creates a number of instances from a software module, and use of the software module is measured according to instances detected at a site of the customer during the monitoring step (Archibald, column 6, lines 48-60; the meter software module generates

accounting information includes many instances the user information, user ID, user account number, etc, if the Archibald's software application is developed using an object-oriented programming such as Sobiski's Java objects, the user information, user ID, user account number, etc in Archibald are objects or instances executed on the customer's computer).

In response to the appellant's argument regarding to Group III: Claim 3, examiner submits that Archibald in modified by Knapton and Sobiski further disclose instances created from a software module are periodically accessed to determine use during the monitoring step (the meter software module accumulates the accounting information for a predetermined period of time, column 12, lines 33-42).

In response to the appellant's argument regarding to Group IV: Claims 4 and 36, examiner submits that Archibald does not teach the monitoring step comprises registering each day that an instance created from a software module is active and wherein the charging step comprises charging the customer a daily rate for use of the software module. However, Ahmad teaches charging the customer a daily rate for use of the software module (column 8, lines 65-67 and column 1, lines 50-51). Therefore, it is obvious to include charging the customer a daily rate for use of the software module in Archibald's for the purpose of time consuming.

In response to the appellant's argument regarding to Group V: Claim 5, examiner submits that Archibald in modified by Knapton and Sobiski further disclose providing a demonstration mode for instances such that instances in the demonstration mode are executable at a customer site without charge (demonstration mode is "trial use field 166", column 9, lines 25-35).

In response to the appellant's argument regarding to Group VI: Claim 6, examiner submits that Archibald in modified by Knapton and Sobiski further disclose maintaining

a single agreement governing use of instances created from the set of software modules for an enterprise (column 7, lines 40-67; Many of instances are created when the user has terminated use of the software module, i.e. an application identification field, a price filed, a user unit field, an amount user field, etc).

In response to the appellant's argument regarding to Group VII: Claim 7, examiner submits that Archibald in modified by Knapton and Sobiski further disclose monitoring a termination date for instances derived from a software module having a time-limited duration (instance "the rent to own field", column 9, lines 19-25).

In response to the appellant's argument regarding to Group VIII: Claims 8, 26, and 41, examiner submits that Archibald does not explicitly teach the self-monitoring process software comprises includes functions for issuing a warning in response to detecting an upcoming expiration date for an instance of a software module or informing the customer of a need to reorder credit to continue using the executable software. However, Ahmad discloses the self-monitoring process software comprises includes functions for issuing a warning in response to detecting an upcoming expiration date for an instance of a software module or informing the customer of a need to reorder credit to continue using the executable software (column 14, line 28-column 15, line 5). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to include these features in Archibald's for the purpose of reminding the customer the expired date of software, so the customer does not forget to re-order credit in order to continue to use the software.

In response to the appellant's argument regarding to Group IX: Claims 12, 32, 33, 74, 75, and 76 examiner submits that Archibald in modified by Knapton and Sobiski further disclose the registered uses of a software module are measured according to

execution and creation of an instance from the software module (see claim 1 above for details).

In response to the appellant's argument regarding to Group X: Claim 14, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the software module is an object class for creating an application engine object or an object class for creating a view engine object (see claim 1 above for details and column 5, line 65-column 6, line 47, the software module is download to the meter module at the consumer computer).

In response to the appellant's argument regarding to Group XI: Claim 15, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the software module is an object class for creating an application engine object or an object class for creating a view engine object (see claim 1 above for details and column 5, line 65-column 6, line 47, the software module is download to the meter module at the consumer computer).

In response to the appellant's argument regarding to Group XII: Claim 16, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the monitoring step comprises determining a time duration that an object instantiated from a software module is active (column 8, lines 43-46).

In response to the appellant's argument regarding to Group XIII: Claims 17, 34, and 77 examiner submits that Archibald in modified by Knapton and Sobiski further disclose the monitoring step comprises registering execution of an instance that tracks throughput of a process (The embedded check data 76 is utilized by the meter module 26 to insure that during operation, the program is being executed as the publisher had intended, column 5, line 65-column 6, line 12).

In response to the appellant's argument regarding to Group XIV: Claim 23, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the reporting step includes identifying the location of an instance created from a software module (an application identification field identifies which software module included in the particular combined digital application module 212, see figure 5 and column 10, lines 1-12).

In response to the appellant's argument regarding to Group XV: Claim 24, examiner submits that Archibald in modified by Knapton and Sobiski further disclose determining that a license manager has not reported to a software brokerage facility and in response registering a communication failure at a central licensing facility (column 19, lines 40-50).

In response to the appellant's argument regarding to Group XVI: Claims 27, 28 and 40, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the software modules relate to industrial manufacturing automation software or relate to industrial manufacturing information software (automation software: active application, information software: passive application column 2, lines 37-44).

In response to the appellant's argument regarding to Group XVII: Claim 30, examiner submits that Knapton discloses providing configuration tools enabling a user to create customized instances from the software modules (column 6, lines 48-67). Therefore, it would have been obvious to one with ordinary skill in the art at the time the invention was made to include this feature in Archibald's for the purpose of helping the user to create customized instances based on the configuration tools in the software product.

In response to the appellant's argument regarding to Group XVIII: Claim 38, examiner submits that Archibald in modified by Knapton and Sobiski further disclose the

authorizing step comprises transmitting a license file containing code enabling use by the customer of the executable software (column 6, lines 33-47 and column 11, line 28-column 12, line 22).

In response to the appellant's argument regarding to Group XIX: Claim 53-55, examiner submits that Archibald discloses a software module facilitating automated distribution of software to customers comprising: a supplier identification; a product description; an executable program; a billing definition includes a usage rate and a lifetime rate (column 6, lines 33-60).

(11) Related Proceedings Appendix

The statement of the related proceedings appendix contained in the brief is correct.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Nga Nguyen

NGA NGUYEN
PRIMARY EXAMINER

Conferees

Abdi Kambiz

Frantzy Poinvil for

LEYDIG, VOIT & MAYER, LTD.
TWO PRUDENTIAL PLAZA, SUITE 4900
180 NORTH STETSON AVENUE
CHICAGO, ILLINOIS 60601-6780